Chapter 21

Data Acquisition System

21.1 Physics Goals

The data acquisition system must read, process, monitor and store the data produced by the various detector systems. Most importantly, the DAQ must provide a distortion-free record of the detector signals resulting from the decay positrons during the 700 μs-long spills from the muon storage ring. Additionally, the system must record all data required to perform the corrections from effects such as pulse pileup, gain instabilities and beam dynamics. Furthermore, the system must allow the monitoring needed to guarantee the overall integrity of data taking and record-keeping needed to document the experimental conditions during data taking.

21.2 Overall Requirements

The overall requirements of the data acquisition system – from the detector sub-systems and the accelerator time structure to the data storage and the data monitoring – are summarized in Table 21.1.

The DAQ must handle the accelerator-defined time structure of the data readout from the detector systems. Under normal operations we anticipate a 12 Hz average rate of muon spills that comprises sequences of four consecutive 700 μs spills with 11 ms spill-separations for each booster batch received by the muon g-2 experiment. The procedures for reading, processing, monitoring and storing these data must not introduce time-dependent losses or time-dependent distortions of the detector signals.

The DAQ must handle the readout, processing, monitoring and storage of the data obtained from the 1296 channels of 800 MSPS, 12-bit, waveform digitizers instrumenting the individual PbF_2 crystals of the twenty four calorimeters. For each spill the raw data will consist of 1296 channels of 700 μs-long streams of continuously-digitized ADC samples. The DAQ must process these raw data into derived datasets including: T method data (*i.e.* individual islands of digitized pulses), Q method data (*i.e.* accumulated histograms of calorimeter spectra), and other calibration, diagnostic and systematic data. At a 12 Hz spill rate the readout (*i.e.* raw) data rate will be about 18 GB/s and the stored (*i.e.* derived) data rate will be about 80 MB/s.

Feature	Requirement
spill time structure	12 Hz spills in groups of 4 spills with 11 ms separations
readout electronics	AMC13-based, spill-async readout and other spill-sync readout
calorimeter raw data	18 GB/s total from 1296 digitizer chans. in 24 μTCA crates
calorimeter derived data	75 MB/s total of T, Q and other derived datasets
other detector data	5 MB/s total of tracker, auxiliary detector raw data
event builder system	spill-based events assembled from ~30 fragments at ~80 MB/s
data monitoring system	online analyser, database to insure data quality / integrity
data storage	transfer to FNAL archive at $\leq$100 MB/s rate for ~2 PB total
run control	web-based local / remote control, monitoring, *etc*

Table 21.1: Summary of the major requirements of the DAQ system.

The DAQ must also handle the readout, processing, monitoring and storage of the data obtained from the three positron tracking stations. This system consists of roughly 3000 channels of straw tubes with associated amplifier-discriminator-TDC electronics. The raw data – consisting of time stamps and spill numbers from individual straw tubes – is expected to yield a r oughly 3 MB/s time-averaged data rate.

Additionally, the DAQ must handle the readout, processing, monitoring and storage of data from the auxiliary detector systems. These systems include: the muon entrance counters, beam position monitors, fiber harp detectors, laser system monitors, electric quadrupole monitors and kicker monitors. The system involves both instrumentation that is operated during normal data taking (*e.g.* the muon entrance detector) and instrumentation that is operated during dedicated data taking (*e.g.* the fiber harp detectors). The read out for the laser calibration monitors will use spare channels of the 800 MSPS, 12-bit, μTCA-based waveform digitizers instrumenting the calorimeters stations. The read out for the muon entrance counters (2 channels), fiber harp detectors (28 channels), electric quadrupole monitors (4 channels) and kicker monitors (6-9 channels) will use a single dedicated crate of 800 MSPS, 12-bit, μTCA-based waveform digitizers. The readout electronics for the beam position monitors will use 1 GSPS CAEN VX1782 waveform digitizers with spill-asynchronous 1 GbE readout.

The expected data rates from auxiliary detector systems are: (i) 17 MB/s during dedicated data taking with the fiber harp detectors and (ii) 3 MB/s during normal data taking with the muon entrance counters, laser calibration monitors, electric quadrupole monitors and kicker monitors.

The DAQ must coordinate the acquisition of data by the frontend readout processes with the accelerator-defined spill cycles. This coordination is designed to incorporate both readout systems where data is transferred synchronously between spill cycles and readout electronics where data is transferred asynchronously with spill cycles (*e.g.* the TCPIP network packets of raw data from μTCA based digitizers and tracker TRMs).

The DAQ must assemble the individual fragments of spill-by-spill data from networked readout processes into complete, deadtime-free records of each muon spill. This includes assembling the data banks of T method and other datasets from the twenty four calorimeter stations as well as the data from the three tracker stations and the auxiliary detector systems. In total the event builder must match and assemble the fragments originating from roughly thirty frontend processes at an expected rate of about 80 MB/s. The resulting fully-assembled spill-by-spill events must be transferred to the Fermilab computing facilities.

The DAQ must provide the local / remote run control for data taking as well as facilities for configuration and readback of configuration parameters such as digitizer settings, multihit TDC settings *etc.* The system must provide the monitoring of data integrity and data quality and a comprehensive database of the experimental conditions and configuration parameters during data taking. The system must additionally provide for local storage of sufficient data for online analysis tasks.

The DAQ will require clean, uninterruptible power of roughly 50 kW total power with appropriate power distribution for roughly thirty rackmount computers and their associated network switches, mass storage devices, *etc.* The control room will require air circulation and cooling power for appropriate temperature and humidity control with temperature, humidity and air velocity sensors with digital readout.

The DAQ will require a reliable, fast network connection between the MC-1 computer room and the Fermilab data storage facilities that is capable of a sustained data rate of roughly 100 MB/s. Based on roughly one year of total running time, the experiment will require a permanent data storage capacity from Fermilab data storage facilities of $\sim$2 Petabytes.

21.3 Recommended Design

21.3.1 DAQ structure

The DAQ will acquire data in blocks that correspond to individual muon spills in the storage ring. Each event will represent a complete deadtime-free history of the entire activity in the detector systems for a complete spill – rather than events corresponding to individual positrons. This scheme will utilize the on-board memories in waveform digitizer and multi-hit TDCs to temporarily buffer the recorded data before its data transfer to the data acquisition. The design will be implemented as a modular, distributed computer system on a parallel, layered array of networked, commodity processors with graphical processing units (GPUs). The DAQ group has developed and operated very similar architectures [1] for the MuLan, MuCap and MuSun experiments at the Paul Scherrer Institute (these experiments involved high statistics, part-per-million measurements of the lifetimes of both free muon and muonic atoms).

The data acquisition system is depicted schematically in Fig. 21.1. It comprises a frontend processor layer responsible for readout and processing of waveform digitizer and multihit TDC data, a backend layer responsible for event assembly and data storage, a slow control layer responsible for control and read-back, and a data analysis layer responsible for monitoring data integrity. The DAQ hardware will comprise a networked cluster of high performance processors running Scientific Linux (4U rackmount eight-core processors with 10 GbE, PCIe3 and NVIDIA K40 GPUs). To maximize the bandwidth, point-to-point networks will handle the traffic between the readout electronics and the frontend process, another sub-network will handle the traffic between the frontend layer and the backend layer, and another network will handle the traffic between the backend layer and the analysis layer. A gateway machine will allow the data transfer between the g-2 private sub-networks and FNAL data storage.

The DAQ software will be based on the MIDAS data acquisition package [2], ROOT data analysis package [3], and NVIDIA's parallel computing platform CUDA [4]. Detailed information on the MIDAS data acquisition package – *e.g.* installation, documentation, user forum, and device drivers for various hardware – are available at Ref. [2]. The MIDAS software consists of function templates and library routines for processes that handle the read out from frontend electronics, event building, data logging DAQ operations. MIDAS also incorporates a fast online database for storing experimental configurations, and a web interface for local / remote control of data taking as well as integrated alarm and slow control systems.

The DAQ will be housed in the computer room in the MC-1 building. 10 GbE optical fiber links ($\sim$30 cables) will provide the point-to-point network connections between the detector sub-systems in the experimental hall and the data acquisition in the control room. for the data readout. A 1 GbE connection between a network switch in the computer room and another network switch in the experimental hall will provide the backbone for IPBus communications between readout electronics and frontend processors.

21.3.2 Asynchronous readout frontends for calorimeter stations

The calorimeter readout consists of one frontend processor per calorimeter station. Each frontend processor will read out the 54 waveform digitizer channels associated with the 6×9 PbF_2 crystals of a single calorimeter station (plus one additional channel for each calorimeter station's laser monitor). Each group of 54 waveform digitizer channels – together with a commercial MCH controller module [5] and a custom-built AMC13 controller module [6] – will occupy a single μTCA crate. The waveform digitizers and AMC13 controller module act as network hosts located on the MCH controller sub-network. The IPBus protocol [7] is used to perform the configuration and control of the waveform digitizers and the AMC13 controller. For read out, the calorimeter frontend process (a TCPIP client) receives digitizer data in 32 kByte blocks from the AMC13 controller (a TCPIP server) via a dedicated 10 GbE point-to-point network connection between the AMC13 controller and the frontend computer.

For each spill the raw calorimeter data will consist of $24 \times 54 = 1296$ channels of 700 μs-long streams of continuously-digitized, 800 MSPS, 12-bit, ADC samples – a total of 1.45 GBytes per spill or 18 GBytes per second (assuming the 12-bit samples each occupy 2 bytes). The frontend readout process comprises: a *TCP_thread* that receives and buffers the raw data blocks from the AMC13 controller, a *GPU_thread* that manages the GPU-based data processing, and a *MIDAS_thread* that handles the transfer of processed data as MIDAS-formatted events to the event builder. Mutual exclusion (Mutex) locks are used to synchronize the execution of the frontend threads. The arrangement was designed

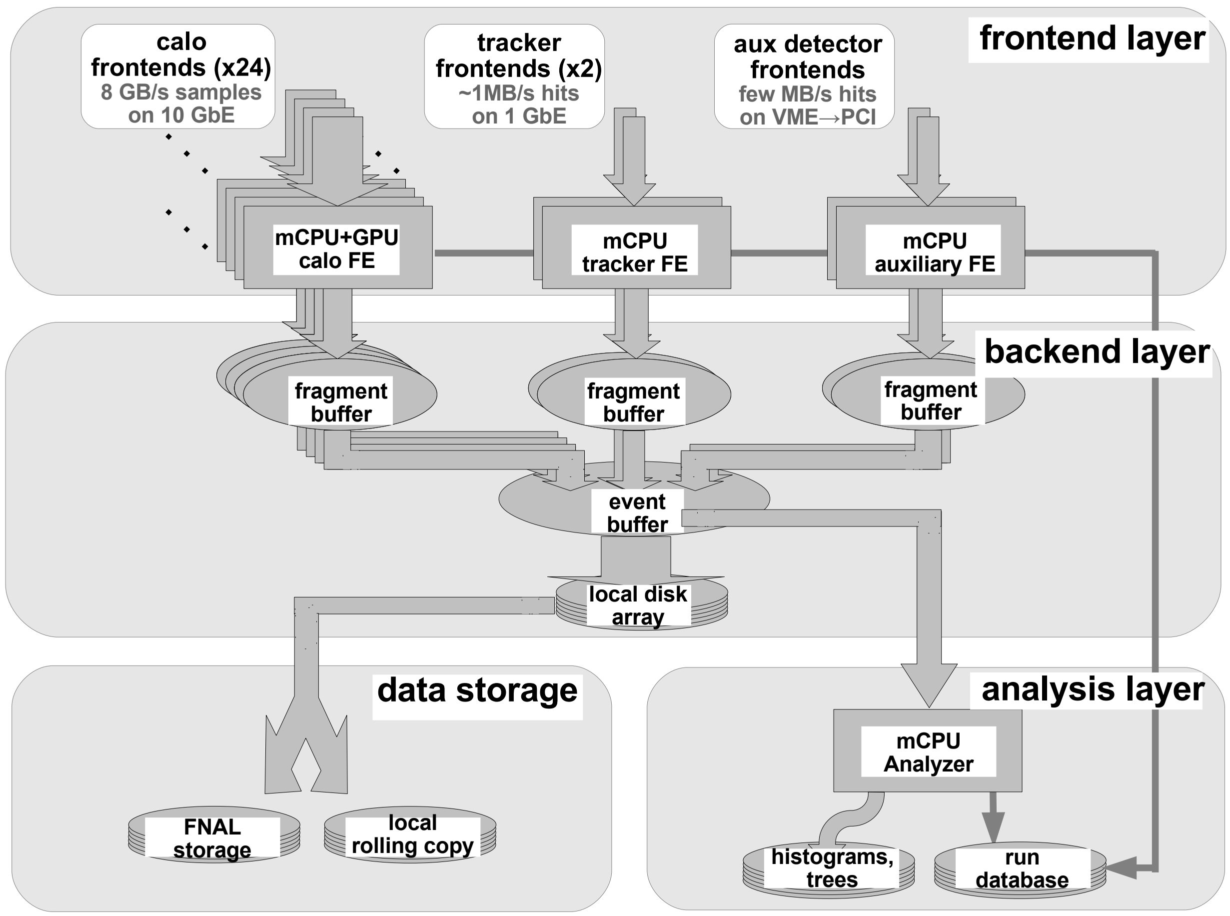


Figure 21.1: Conceptual design of the g-2 data acquisition. The figure shows: (i) the frontend layer for readout and processing of data from the calorimeter stations, tracking stations and auxiliary detector systems, (ii) the backend layer for event building and data migration, (iii) the analysis layer for monitoring data quality and recording run-by-run experimental conditions, configuration parameters, *etc.*, and (iv) data storage. The layers comprise arrays of networked commodity processors.

to provide the necessary performance to compress the continuously-digitized ADC samples into the T/Q method datasets at the software level. The T method datasets will consist of individual "islands" of above-threshold calorimeter signals and the Q method datasets will consist of sum histograms of consecutive spills of continuously digitized samples. Additionally – for activities such as detector commissioning and data monitoring – the design allows for storage of either all raw data (at low raw data rates) or pre-scaled raw data (at high raw data rates).

The algorithms for constructing the T/Q method datasets involve copying, masking and summing arrays and basic digital signal processing and therefore are ideally suited to GPU-based parallelization using standard algorithms and CUDA libraries. A major advantage of GPU-based processing of raw data into compressed datasets is the comparative ease of testing and implementing a variety of algorithms for processing sub-tasks such as digital filtering, pulse triggering, pulse clustering, *etc.* The scheme also offers the flexibility to implement other datasets – such as pile-up datasets (*e.g.* by summing fills before storing islands) and diagnostic datasets (*e.g.* by storing prescaled fills of continuously digitized samples) – as needed.

21.3.3 Asynchronous readout frontends for tracker stations

The positron tracking system consists of three tracking stations that each comprise about 1000 individual channels of straw tube detectors. The raw data from the frontend electronics of the three tracker stations are transferred via a serial link to custom-built tracker readout modules (TRMs). The TRMs – and their associated MCH and AMC13 controllers – are housed in a single μTCA crate. The TRM modules act as network hosts on the MCH controller sub-network and are configured and controlled using the IPBus protocol via the 1 GbE network connection between the MCH controller and the frontend computer. The tracker readout process (a TCPIP client) receives TRM data in 32 kByte blocks from the AMC13 controller (a TCPIP server) via a dedicated 10 GbE point-to-point network connection between the AMC13 controller and the frontend computer.

Each tracker frontend process will receive the TRM raw data – *i.e.* tracker hits defined by a channel number, spill number and time stamp – and pack and dispatch these data as MIDAS-formatted databanks over the frontend network to the event builder. Since the readout mechanisms for the μTCA-based WFD and TRM modules are the same, the same readout framework can handle the WFD data readout and TRM data readout. The TRM / WFD frontend software will only differ in details related to the device-dependent IPBus communication and associated online database entries and the presence (or the absence) of the GPU-based data processing for the WFD (TRM) data.

21.3.4 Synchronous readout frontends for auxiliary detectors

The auxiliary detector systems comprise the muon entrance counters, beam position monitors, fiber harp detectors, electric quadrupole monitors and kicker monitors. The beam position monitors will be readout by 1 GSPS CAEN VX1742 waveform digitizers. The other auxiliary detectors with be readout by a dedicated μTCA crate of 800 MSPS, 12-bit waveform digitizers. In both cases the data are trasferred over 11/10 GbE ethernet and readout using the same framework as the calorimeter frontend and the tracker frontend. The calorimiter, tracker and auxiliary readout frontends will differ only in the details of the IPBUS configuration and the corresponding ODB structure for the relevant WFD /TDC modules.

21.3.5 Master frontend and accelerator-DAQ synchronization

The DAQ design incorporates a master frontend process and hardware control logic in order to synchronize the data acquisition readout cycles with the accelerator-defined spill cycles. Importantly, the master frontend and control logic must accommodate both the spill-synchronous data readout from the VME crates via Struck SIS 3100/3300 interface modules

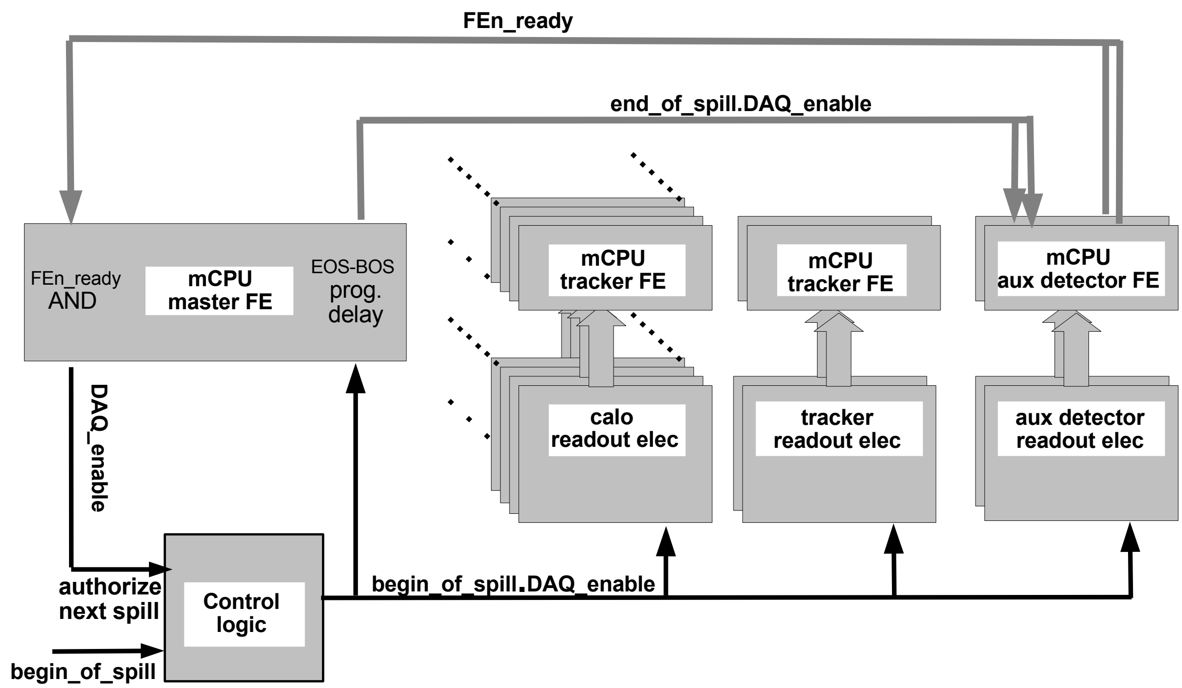


Figure 21.2: Conceptual design of the DAQ control logic. The logic is based upon an accelerator-defined *begin_of_spill* signal and a DAQ-defined *DAQ_enable signal*. When the DAQ is ready for data taking, it enables the distribution of the next *begin_of_spill* to the WFD/TDC electronics via a *DAQ_enable signal*. When the any synchronous readout processes have completed their required tasks between successive spills they report their readiness to the master frontend process via a *FE_ready* remote procedure call. The master frontend issues the next *DAQ_enable* signal when all synchronous frontends have reported their readiness.

and the spill-asynchronous data readout from the μTCA crates via 10 GbE network links. The hardware control logic design uses a *DAQ_ready* signal to authorize the recording of the data from the next spill by the WFD/TDC electronics. The software control logic uses remote procedure calls (RPCs) to provide status messages (*i.e. DAQ_ready*, *begin_of_spill*, *end_of_spill*) to readout processes and *FE_ready* messages from the synchronous frontends to the master frontend.

The design of the synchronization logic that coordinates the spill cycles and readout cycles is shown in Fig. 21.2. The design involves two hardware control signals – an accelerator-derived *begin_of_spill* signal and DAQ-derived *DAQ_enable* signal. On starting a run, the master frontend processes issues a *DAQ_enable* signal to authorize recording the next spill. The subsequent authorized *begin_of_spill* signal, *i.e. begin_of_spill·DAQ_enable*, is then distributed to both the WFD/TDC electronics to enable data recording and the master frontend to generate *begin_of_spill* and time-delayed *end_of_spill* RPCs. The digitization time intervals for each detector system is a configuration parameter that is stored in the DAQ database and set in the readout electronics. The synchronous readout processes – which require the readout of the data size before the acquisition of the next spill – use remote procedure calls to report their readiness for data taking. This "readiness" report is made by a *FE_ready* remote procedure call from the synchronous readout processes to the master process. On receipt of all *FE_ready* RPCs from all synchronous readout processes, the master processes then issues another *DAQ_enable* signal to authorize recording the next spill. The asynchronous readout processes – that receive TCPIP data packets over 10GbE network links with AMC13 modules – are not required to report their readiness before acquiring another spill. [1]

Note we plan to use a PCI-based GPS synchronization card [9] to GPS timestamp the digitized spills in order to facilitate later coordination between the detector system readout and the magnetic field readout.

21.3.6 Event building and data logging

Each frontend readout process will transmit its spill-by-spill data fragments as MIDAS-format databanks across the frontend network to the backend processor. Initially, the data fragments from the twenty four calorimeter processes, three tracking system processes and various auxiliary detector processes, are transferred to shared memory segments on the backend machine (one memory segment per frontend process). After matching the MIDAS serial numbers and muon spill indexes of event fragments the event builder process assembles all data fragments into single events representing a complete record of each spill. The spill events are then written by the event builder process to a further memory segment known as the system memory segment and are then available for data storage tasks, data analysis tasks, *etc.*

The backend layer will use a two step procedure to permanently store a full copy of the data on the Fermilab Computing facilities and temporarily store a rolling copy of the recent data on our analysis layer. First, the data will be transferred from the system memory segment to a temporary disk file on a local redundant disk array on the backend processor. Next, the temporary data files on the backend processor will be asynchronously migrated to both the Fermilab computing facilities for permanent storage and the DAQ analysis layer for local analysis projects. This approach will minimize any delays in data taking due to latencies associated with the permanent archiving of the experimental data and make the current data available for local analysis and monitoring projects.

21.3.7 Online database and data monitoring

The DAQ analysis layer will provide both integrity checking and online histogramming. The online analyzer will receive events over the network from the system memory segment on

[1]The asynchronous readout processes use an independent readout thread with a ring buffer to maximize the TCPIP data transfer rate.

the backend layer. These events will be received "as available" in order to avoid introducing any delays into the readout or the data storage. The online analysis will utilize the ROME framework generator for event-based data analysis [11] In ROME the analysis framework is automatically generated using a XML framework definition file that defines all necessary experiment-specific classes.

The analyzer will utilize a modular, multistage approach to analysis tasks where different analysis tasks will be implemented as individual analyzer modules and then switched on / off as needed. Each analysis module will have access to a global structure that contains both the raw MIDAS databanks from the readout processes and any derived MIDAS databanks from the preceding analysis modules. Low-level modules will be responsible for unpacking the databanks and checking their integrity. Intermediate-level modules will be responsible for various histogramming tasks to ensure the correct operations of detector systems. High-level modules will be responsible for online "physics" analysis such as fits to the precession signal.

The data acquisition system will incorporate database support to provide a comprehensive run-by-run record of the experimental conditions, configuration parameters, *etc.*, during the entire experiment.[2] The run-by-run database will store information derived from the MIDAS online database such as run start time, run stop time, operator run-time comments, the number of events, and hardware settings including the high voltage setting, digitizer configuration parameters, multihit-TDC configuration parameters, *etc.* In addition, the database will record such quantities as detector gains, pedestals, *etc.*, fitted frequencies, lifetimes, *etc.*, that are derived from the analysis layer. This information are the foundations for the offline data analysis.

A $g-2$ web-based interface will provide both local and remote access to run-by-run histograms, trend plots and the experimental database.

21.4 Design Performance

21.4.1 Test stands for prototyping and development

Several test stands are being used for the prototyping work on the data acquisition system. A test stand at UKy and another stand at Fermilab are being used for the development of the calorimeter readout system. A test stand at UCL is supporting the development work on the tracker readout system and another stand at Dubna is supporting the development of the ROME data monitor.

As an example, the UKy test stand that was used for several R&D projects on the calorimeter readout, is shown in Figs. 21.3 and 21.4. It comprises a network of three frontend processors and one backend processors. The backend processor hosts the MIDAS server process (MSERVER) that manages inter-process communications, the MIDAS web daemon (MHTTPD) that provides run control, the MIDAS event builder (MEVB), as well

[2]The MIDAS data acquisition package includes a central database called the "Online DataBase" (ODB). The ODB stores run parameters, frontend readout parameters, backend logging parameters, status / performance data, as well as other user-defined information. All processes participating in data taking have full access to the information in the MIDAS ODB.

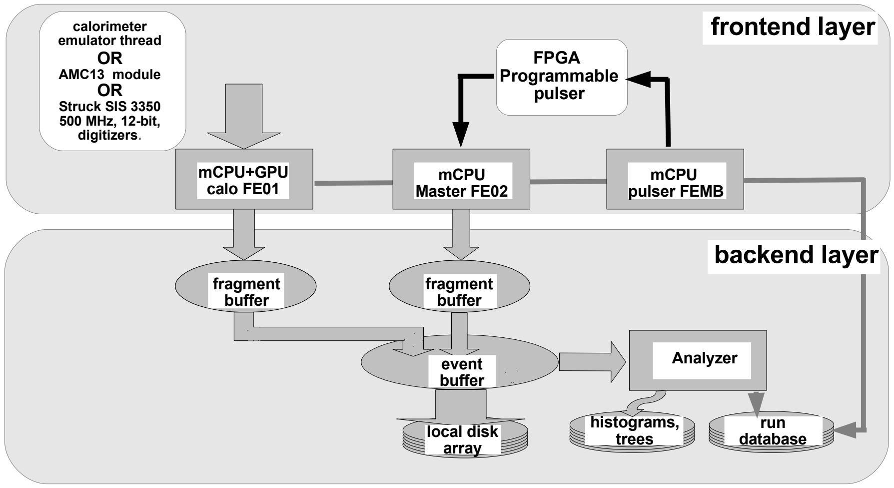


Figure 21.3: Layout of UKy MIDAS-based DAQ-platform for R&D projects. The frontend FE01 incorporates both readout of emulated continuously-digitized ADC samples from a calorimeter simulation process and real continuously-digitized ADC samples from a Struck SIS3350 digitizer system. The frontend FEMB emulates the Fermilab accelerator control signals and the frontend FE02 provides the software synchronization of spill-by-spill readout. The system also includes an event builder layer and data monitor layer.

Figure 21.4: Photograph of the DAQ test stand including μTCA-based electronics and VME-based electronics at UKy.

as data storage and analysis tools. Frontend processor FE01 comprised two quad-core Intel Xeon X5550 CPUs and a 2496-core, NVIDIA Kepler K20 GPU. As is planned for the full DAQ system, the test stand was arranged as a local sub-network of frontend processors with a backend file server.

The Fermilab, UCL and UKy test stands all incorporate both prototype readout electronics as well as either calorimeter station emulators (Fermilab / UKy) or tracker station emulators (UCL).

21.4.2 Development and prototyping of T / Q method data processing

Fig. 21.5 shows a typical spill of simulated data – *i.e.* $3.5{\times}10^5$ ADC samples of 700 μs continuous-digitization – that was Monte-Carlo generated by the calorimeter emulator. In building streams of ADC samples, the decay positrons were generated with the appropriate energy-time distributions for 3.094 GeV/c decays and the calorimeter hits were generated with appropriate x-y distributions and pulse shapes. The raw data were read out and processed into T/Q method datasets in the DAQ frontend layer and then analyzed and histogrammed in the DAQ analysis layer. Representative plots of T method energy and time distributions of decay positrons are shown in Fig. 21.6.

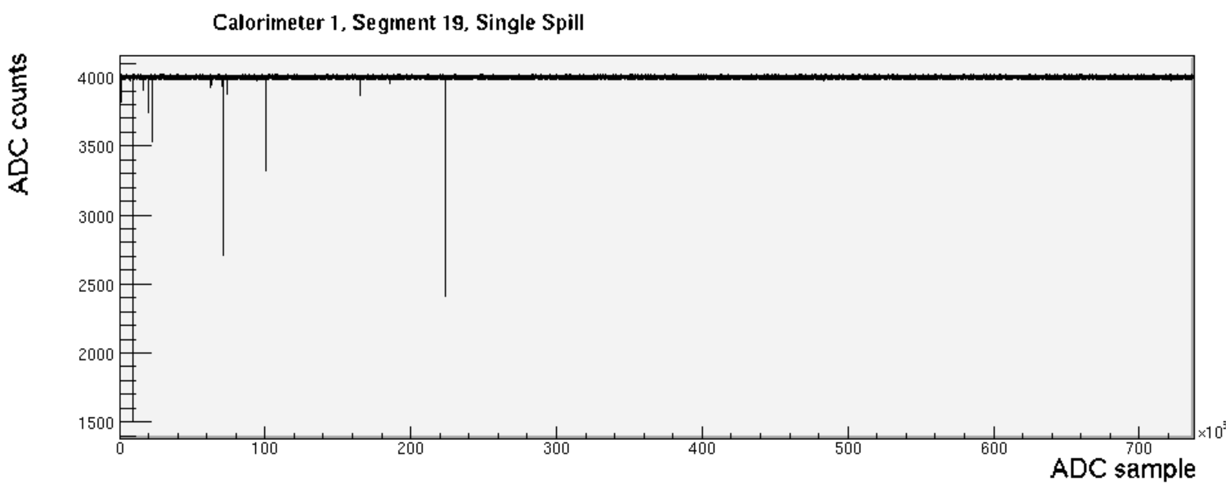


Figure 21.5: A representative single spill of simulated data generated by the calorimeter emulator. The data correspond to $3.5{\times}10^5$ ADC samples of 700 μs continuous-digitization for one segment of one calorimeter.

Results from frontend timing tests of the GPU-based, T/Q method processing of the simulated calorimeter data are shown in Fig. 21.7. After completing the readout of each spill of ADC samples the raw data are transferred from the CPU memory to the GPU memory. The GPU then initiates a sequence that involves: derivation of the segment-summed calorimeter samples from the individual crystal segment samples, identification of the T method above-threshold pulses in the summed calorimeter samples, assembly islands of the T method above-threshold islands with pre-/post-samples, and transfer of the resulting

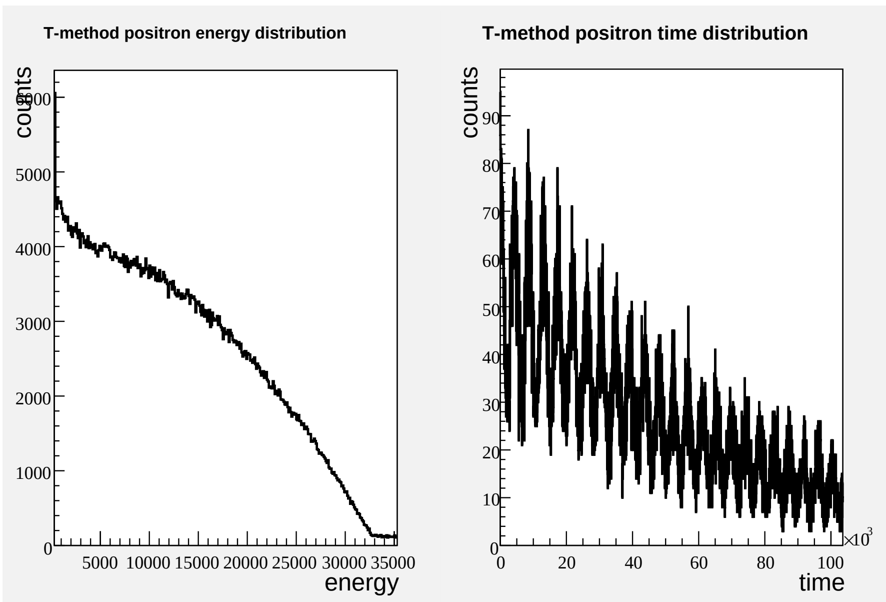


Figure 21.6: Energy distribution (lefthand plot) and time distribution (righthand plot) of energy / times of positron hits. The data were generated as continuous-digitization spills by the calorimeter emulator, were processed into T Method datasets in the calorimeter, and histogrammed in the analysis layer. The energy distribution shows the positron endpoint energy and the time distribution shows the anomalous precession frequency.

T method data from the GPU memory to the CPU memory. Additionally, a Q method dataset was constructed by summing consecutive blocks of 32 ADC samples of digitizer data and then copied from the GPU memory to the CPU memory. Finally, the T/Q method datasets are packaged into MIDAS databanks and transferred to the backend layer.

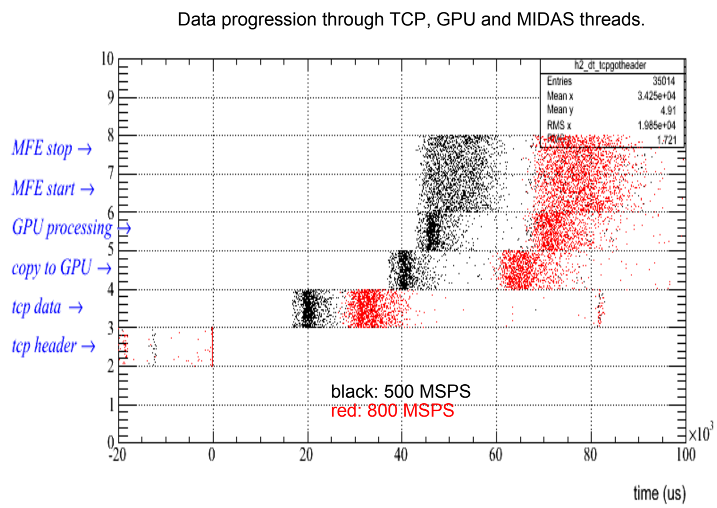


Figure 21.7: Timing plot of the readout, processing and transmission of simulated data through the TCP, GPU and MIDAS threads in the calorimeter readout frontend. The *TCP_thread* handles the receiving and unpacking of the 10 GbE network data, the *GPU_thread* manages the GPU processing into T, Q and other datasets, and *MFE_thread* manages the assemble and transfer of MIDAS-formatted events. The plot show the processing time for spills digitized at both 500 MSPS and 800 MSPS. The above results were obtained using the UKy test stand.

21.4.3 Development and prototyping of event building

Also conducted were timing tests of event building on simulated databanks in the backend layer of the data acquisition. The tests showed the rate limitations in event building were largely governed by memory copy operations during event fragment assembly. For the backend processor in the UKy test stand – a six core, Intel i7 processor with 8 GBytes of high-bandwidth DDR3 memory – the system was able to handle a data rate exceeding 100 MB/s without introducing time delays.

21.4.4 Development and prototyping of AMC13 readout

Both the UKy test stand and the Fermilab test stand include a μTCA crate, commercial MCH controller module and custom-built AMC13 module. Since April 2015 the Fermilab test stand also incorporates a prototype 5-channel waveform digitizer.

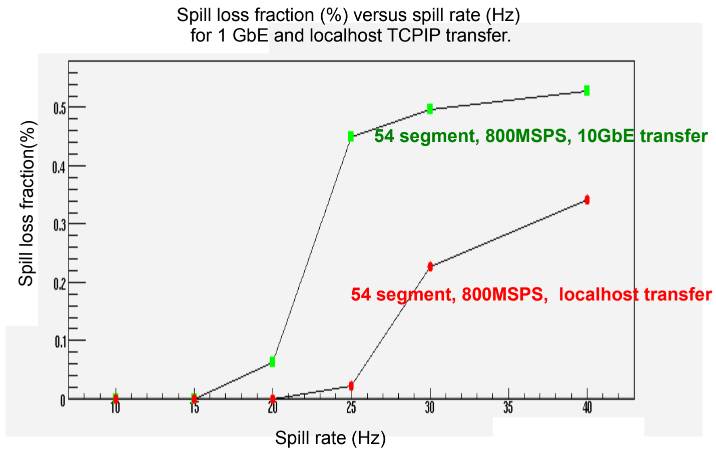
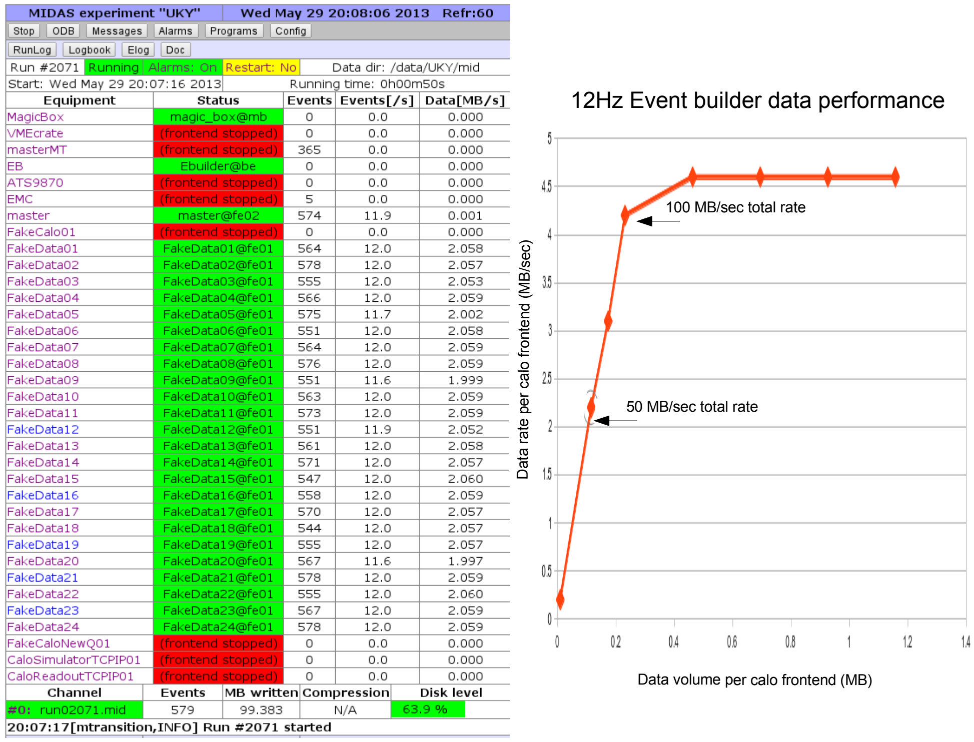


Figure 21.8: Spill lost fraction (%) versus spill rate (Hz) for 800 MSPS digitization with TCPIP data transfer over 10 GbE (green) and localhost (red).

This hardware has enabled the development work on the control, configuration and readout of AMC13 modules and more recently the control and configuration of the waveform digitiizer module. To date we've developed a *CaloReadoutAMC* MIDAS frontend for the readout of the AMC13 events and a *CaloTriggerAMC* MIDAS frontend to trigger the AMC13 to transmit events to the readout process. The new software incorporates both the IPbus protocol for AMC13 configuration / control and a TCPIP thread for AMC13 10 GbE readout. The TCPIP thread incorporates a time-efficient algorithm for unpacking the 32 kB blocks of AMC13 aggregated data into per-channel, per-module, time-ordered ADC samples. We have successfully demonstrated both the IPbus communication for AMC13 configuration / control and the 10 GbE readout of the AMC13 aggregated data.

21.5 Alternatives

The collaboration considered two alternative data acquisition frameworks: the CODA DAQ package [12] that is used at JLab and the artdaq DAQ package [13] that is under development at Fermilab. One advantage of using the MIDAS is that collaboration members have already developed very similar DAQ architectures with the MIDAS framework for other experiments. The g-2 DAQ can therefore profit from the software/hardware development for the earlier experiments. Another advantage of MIDAS is the availability of an extensive range of DAQ tools including an event builder, an analysis framework, a slow control system, a data alarm system, data storage and database tools, as well as large collections of device drivers for readout hardware. Its modularity makes MIDAS very convenient for both data acquisition development and detector system prototyping. MIDAS has an active community of software



MIDAS experiment "UKY"		Wed May 29 20:08:06 2013 Refr:60		
Stop ODB Messages Alarms Programs Config				
RunLog Logbook Elog Doc				
Run #2071 Running Alarms: On	Restart: No	Data dir: /data/UKY/mid		
Start: Wed May 29 20:07:16 2013		Running time: 0h00m50s		
Equipment	**Status**	**Events**	**Events[/s]**	**Data[MB/s]**
MagicBox	magic_box@mb	0	0.0	0.000
VMEcrate	(frontend stopped)	0	0.0	0.000
masterMT	(frontend stopped)	365	0.0	0.000
EB	Ebuilder@be	0	0.0	0.000
ATS9870	(frontend stopped)	0	0.0	0.000
EMC	(frontend stopped)	5	0.0	0.000
master	master@fe02	574	11.9	0.001
FakeCalo01	(frontend stopped)	0	0.0	0.000
FakeData01	FakeData01@fe01	564	12.0	2.058
FakeData02	FakeData02@fe01	578	12.0	2.057
FakeData03	FakeData03@fe01	555	12.0	2.053
FakeData04	FakeData04@fe01	566	12.0	2.059
FakeData05	FakeData05@fe01	575	11.7	2.002
FakeData06	FakeData06@fe01	551	12.0	2.058
FakeData07	FakeData07@fe01	564	12.0	2.059
FakeData08	FakeData08@fe01	576	12.0	2.059
FakeData09	FakeData09@fe01	551	11.6	1.999
FakeData10	FakeData10@fe01	563	12.0	2.059
FakeData11	FakeData11@fe01	573	12.0	2.059
FakeData12	FakeData12@fe01	551	11.9	2.052
FakeData13	FakeData13@fe01	561	12.0	2.058
FakeData14	FakeData14@fe01	571	12.0	2.057
FakeData15	FakeData15@fe01	547	12.0	2.060
FakeData16	FakeData16@fe01	558	12.0	2.059
FakeData17	FakeData17@fe01	570	12.0	2.057
FakeData18	FakeData18@fe01	544	12.0	2.057
FakeData19	FakeData19@fe01	555	12.0	2.057
FakeData20	FakeData20@fe01	567	11.6	1.997
FakeData21	FakeData21@fe01	578	12.0	2.059
FakeData22	FakeData22@fe01	555	12.0	2.060
FakeData23	FakeData23@fe01	567	12.0	2.059
FakeData24	FakeData24@fe01	578	12.0	2.059
FakeCaloNewQ01	(frontend stopped)	0	0.0	0.000
CaloSimulatorTCPIP01	(frontend stopped)	0	0.0	0.000
CaloReadoutTCPIP01	(frontend stopped)	0	0.0	0.000
Channel	**Events**	**MB written**	**Compression**	**Disk level**
#0: run02071.mid	579	99.383	N/A	63.9 %
20:07:17[mtransition,INFO] Run #2071 started				

Figure 21.9: Test of event building for twenty four networked frontends. Lefthand side is screen shot of MIDAS control page show twenty four calorimeter frontends, master frontend (handling control logic), magic box frontend (generating spill cycles), and event builder and data logger. Righthand side show frontend data rate (MB/sec and total event builder rate (MB/sec) as function of data size of processed spills.

developers and is widely used at numerous nuclear and particle physics laboratories.

Several alternatives were considered for processing the raw calorimeter data ino T, Q and other method datasets. In particular, the collaboration considered the possibilities of deriving the T/Q method datasets at either the software level in the frontend processors or the firmware level in the digitizer FPGA electronics. One of the major advantages of the software implementation is the greater flexibility to modify or add new datasets or analysis algorithms as needed during the design, commissioning and running phases of the experiment. If necessary, one advantage of implementing some tasks in firmware is the lowering of the raw data rates from the readout electronics to the frontend processors.

Different architectures and parallelization schemes were also considered for the calorimeter frontend processors. In one parallelization scheme using multicore CPUs the frontends process individual spills in separate CPU threads in order to achieve the necessary data compression bandwidth. In another parallelization scheme using many core GPUs the frontends process individual samples in separate GPU threads in order to achieve the necessary data compression bandwidth. In tests, the GPU-based approach was found better matched to parallelizing the tasks involved in deriving T/Q method datasets, and takes advantage of general purpose algorithms and CUDA library functions for such operations.

One alternative under consideration for the data acquisition from the auxiliary detectors is the use of extra channels of μTCA-base waveform digitizers in place of existing channels of VME-based waveform digitizers. This alternative would reduce the complexity of the DAQ system through the use of a single readout interface.

21.6 ES&H

The components of the data acquisition do not involve either hazardous materials or unusual electrical / mechanical hazards. The system will comply with safety standards for power distribution, will require appropriate cooling power for the $\leq$50 kW computer system, and will require temperature, humidity and air velocity sensors.

21.7 Risks

Since the calorimeter readout involves the computation of derived T/Q method datasets from continuously digitized ADC samples, the largest risk in the data acquisition system is the corruption or the distortion of the positron time spectra. These risks are mitigated by DAQ integrity testing using DAQ test stands (see Sec. 21.8). They are further mitigated by pre-scaled collection of raw data and continuous monitoring of data integrity during experiment running.

A further risk is insufficient performance of the data acquisition – in particular the derivation of T/Q method datasets – that would impact rates of data accumulation during data taking. This risk is mitigated by the possibility of moving some tasks in the dataset derivation from the GPU hardware to the Kintex 7 FPGA in the AMC13 interface module. The risk is also mitigated by the flexibility of the GPU processing and the ease of a GPU upgrade. In addition, the combined memory of the digitizer modules and the AMC13 interface module, is able to buffer about 13 seconds of consecutive spills and therefore help to mitigate effects of rate variations due to the irregular fill cycle.

Another risk is delays in the software development of the data acquisition that would impact the schedules for detector installation, experiment commissioning and data taking. The DAQ is assembled from commodity computing hardware, so procurement and delivery is not likely a significant risk. The staged development and release approach to DAQ work should reduce the risk of delays in the availability of the software.

21.8 Quality Assurance

Fermilab, UCL and UKy have each established test stand for DAQ development, testing and quality assurance. The first stage of development and implementation for the calorimeter data readout, data processing and event building has been underway for two years using a calorimeter station emulator. The second stage of development and implementation was begun in Spring 2014 with the acquisition of a μTCA crate and MCH / AMC13 control modules. Other collaborators at Cornell and Washington are also establishing test stands for DAQ-related tasks on detector prototyping and electronics readout.

Quality assurance of the DAQ components will also be conducted using the test stands at various institutions. The major DAQ specifications requiring quality assurance are Such stress tests of DAQ components include: (i) rate, integrity and functionality tests of 10 GbE data transfer from the AMC13 controllerr and the frontend TCPIP readout thread, (ii) rate, integrity and functionality tests of PCIe data transfer and GPU data processing of the raw calorimeter samples into the derived data datasets in the frontend GPU thread, (iii) functionality and integrity tests of the IPBus control of the AMC13, Rider modules, and TRM modules, (iv) functionality and integrity tests of the data readout for the tracker stations and the auxiliary detectors, (v) functionality and integrity tests of the framework for the flexible readout of spill-asynchronous and spill-synchronous frontends, (vi) functionality and integrity tests of the framework for the DAQ / electronics control involving the accelerator generated begin-of-spill signal and the DAQ generated spill-authorization signal. (vii) rate, integrity and functionality tests of event builder system to handle approximately 30 event-fragments and a $\leq$100 MB/s rate, (viii) integrity and functionality tests of event builder system to handle both fragments for event building and raw / histogram data-types without event building. The data quality monitor is an integral part of the quality assurance for these parameters.

A phased build approach will be utilized for the DAQ implementation in the MC-1 computer room. As of May 2015 a system for readout and processing of two calorimter stations is installed in Fermilab MC-1. Our schedule is to complete a 6 calorimeter readout system during Summer 2015, a 12 calorimeter readout system during Fall / Winter 2015, and a 24 calorimeter readout system during Spring / Summer 2016. Mock data taking with the laser calibration system will begin Summer 2016.

21.9 Value Management

Significant value engineering has been employed in the design of the DAQ as is demonstrated in the alternatives section. In the future, we will continue to monitor developments in commodity electronics for data processing, networking, storage, *etc.* We plan to make our purchases at appropriate times to minimize costs and maximize performance in meeting the DAQ requirements for the experiment.

In addition to T/Q method data compression we plan to evaluate the lossless compression of digitizer data using standard libraries (*e.g.* ZLIB library [10]). For continuously-digitized ADC samples that consist of occasional pulses the loss-less algorithms should offer efficient compression. However, the sensitivity of compression to sources of noise, as well as the time-cost of the data compression, must be carefully studied.

21.10 R&D

Our R&D – discussed in Sec. 21.4 – will continue using the DAQ test stands at Fermilab, UCLm UKy and other institutions. Future R&D projects include:

- continuing development, testing and implementation of the data readout from the AMC13 controller with particular attention to optimizing performance and integrity testing

- continuing development, testing and implementation of the GPU-based processing of the T, Q, and other datasets using emulator calorimeter data and waveform digitizer data
- further development, testing and implementation of the hardware / software logic for the synchronization of the accelerator and readout cycles for synchronous and asynchronous frontends
- further development, testing and implementation of the event building for synchronous / asynchronous data sources.

References

[1] V. Tishchenko, S. Battu, S. Cheekatmalla, D. B. Chitwood, S. Dhamija, T. P. Gorringe, F. Gray and K. R. Lynch *et al.*, Nucl. Instrum. Meth. A **592**, 114 (2008) [arXiv:0802.1029 [nucl-ex]].

[2] https://www.triumf.info/wiki/DAQwiki/index.php/MIDAS (TRIUMF MIDAS homepage) and http://midas.psi.ch/. (PSI MIDAS homepage). Also see: E. Frlez *et al.*, Nucl. Instrum. Meth. **A526** (2004) 300;

[3] Rene Brun and Fons Rademakers, Nucl. Instrum. Meth. **A389** (1997) 81; see also http://root.cern.ch/.

[4] Jason Sanders and Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison-Wesley, ISBN-10: 0131387685. Also http://docs.nvidia.com/cuda/index.html

[5] V. Rose Microsystems, VRM-MCH μTCA Carrier Hub, http://www.vrosemicrosystems.com/product-159.htm

[6] Eric Hazen, CMS AMC13 hub card, http://joule.bu.edu/ hazen/CMS/AMC13/ AMC13_Short_Spec_05Oct2012.pdf

[7] IPBus Software Wuite link at https://svnweb.cern.ch/trac/cactus

[8] Struck Innovative System, Harksheider Str. 102A, 22399 Hamburg, Germany; see also http://www.struck.de/.

[9] Spectracom Corporation, 1565 Jefferson Road, Suite 460, Rochester, NY 14623, . http://www.spectracomcorp.com/ProductsServices/TimingSynchronization.

[10] D. A. Huffman, Proceedings of the Inst. of Radio Engineers, **40** (1952) 1098; see also http://www.zlib.net/.

[11] ROME framework generator for event-based data analysis. https://midas.psi.ch/rome/

[12] Jefferson Laboratory Data Acquisition Group CODA Wiki, https://coda.jlab.org/wiki/index.php/Main_Page

[13] Kurt Biery, artdaq: An event filtering framework for fermilab experiments, Real Time Conference (RT), 2012 18th IEEE-NPSS, 9-15 June 2012,